DYNAMICSCON PRESENTATION

DYNAMICSCON LIVE

POWERED BY DUG

SEPTEMBER 14-16,
2022

How to be a Bada$$ BA

Melissa Erickson

- Team Manager
- melissae@stoneridgesoftware.com

Agenda

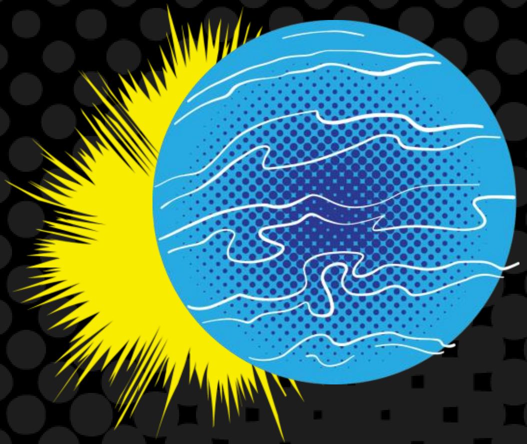
- What is a BA?
- Why is a BA Important?
- BA Role in a Project
- BA Toolkit
- BA Superpowers
- How to Create Bada$$ Requirements
- BA Top Tips

Stats

> "It will be no surprise if a large, established company fails in the coming years because of an out-of-control IT project."
>
> –Harvard Business Review

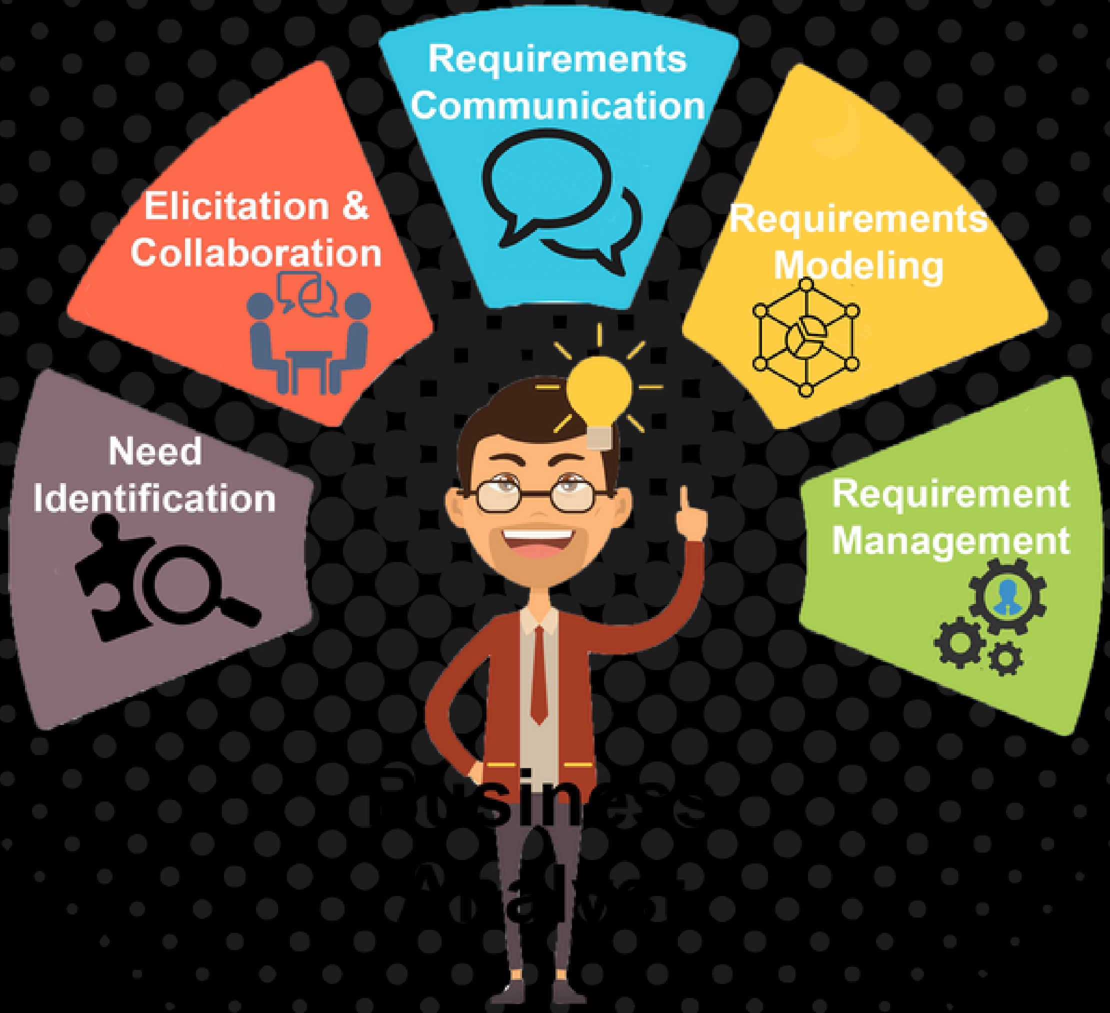
- Average project cost overrun is **27%**
- ***64%*** *of projects meet their goals*
- ***70%*** *of companies report having at least one failed project in the last year*
- *Organizations lose* **$109 million** *for every* **$1 billion** *invested in projects and programs*
- *High-performing organizations successfully complete* **89%** *of projects*
- *Low performers only complete* **36%** *successfully*
- *Only* ***one-third*** *of companies always prepare a business case for new projects*

Business Analyst



A Business Analyst (known as a BA) is defined as a person who analyzes and documents processes or systems of businesses. The typical role focuses on defining business requirements.

What is a BA?



Need Identification

Elicitation & Collaboration

Requirements Communication

Requirements Modeling

Requirement Management

Innovia Consulting

Ingram Micro Cloud

SPS Commerce

Own {backup}

DynamicsCon Live

Why is a BA Important?

ROI – (Total value) minus (Total cost)

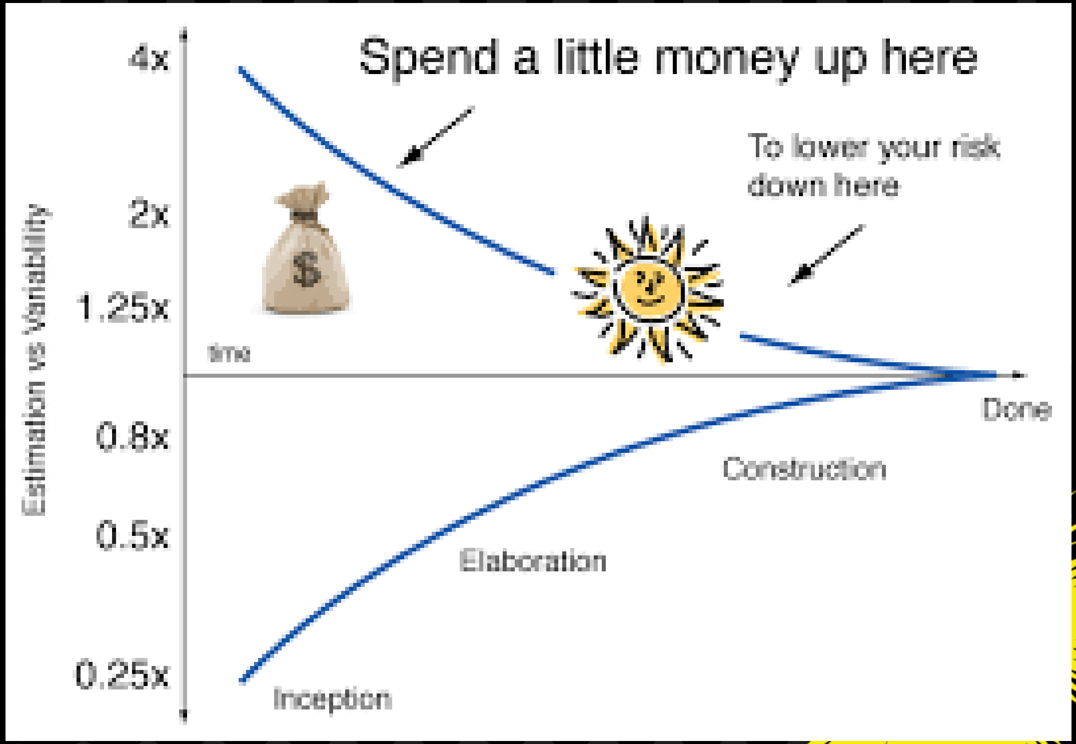
Increase Total Value

1. Prioritizing Requirements
2. Discovering new business needs
3. Increasing communication
4. Provide a framework for scaling
5. Support User adoption

Decrease Total Cost

1. Documenting the right requirements
2. Reduction in rework
3. Shorten the project length
4. Reduction in stakeholder time
5. Discover more cost-effective solutions

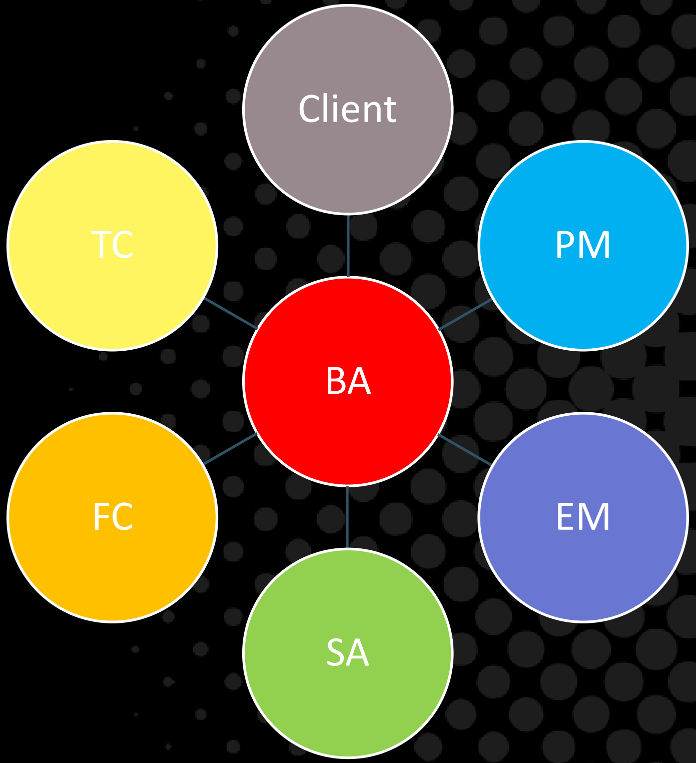
Cone of Uncertainty



Spend a little money up here

To lower your risk down here

BA Role in a Project



Models Business (As-Is) and Requirements (To-Be)

Gathers & Documents Requirements

Develops Requirements Work Plan

Manages Changing Requirements

Validates Requirements

BA Toolkit

DevOps, JIRA, Excel, OneNote

- Use cases, requirements, acceptance criteria, test cases

Teams

- Meetings, recordings

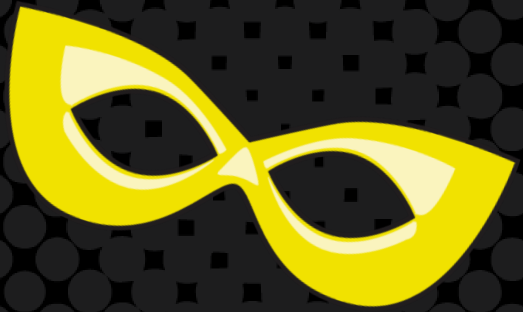
SharePoint

- Store recordings, documentation

Visio

- Workflows, use case diagrams

What do you think are the 3 superpowers?



Innovia CONSULTING

INGRAM MICRO CLOUD

SPS COMMERCE

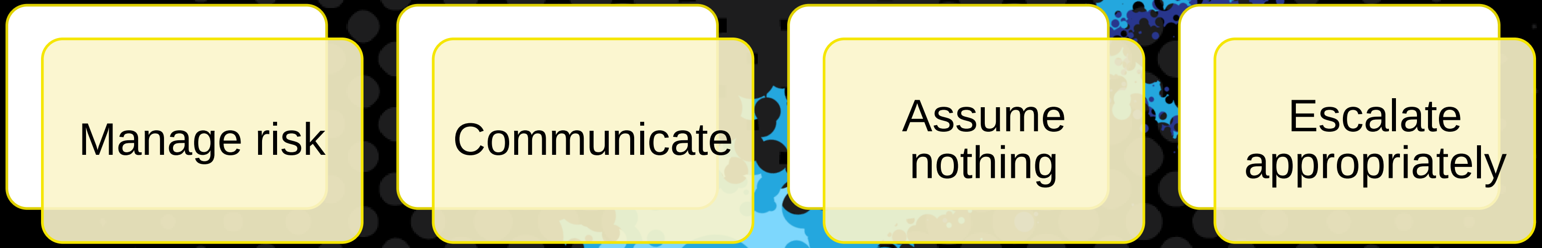
Own {backup}

DYNAMICSCON LIVE

POWERED BY DUG

BA Superpowers

- Documenting and Communicating Risks



Manage risk

Communicate

Assume nothing

Escalate appropriately

BA Superpowers

Solving Problems

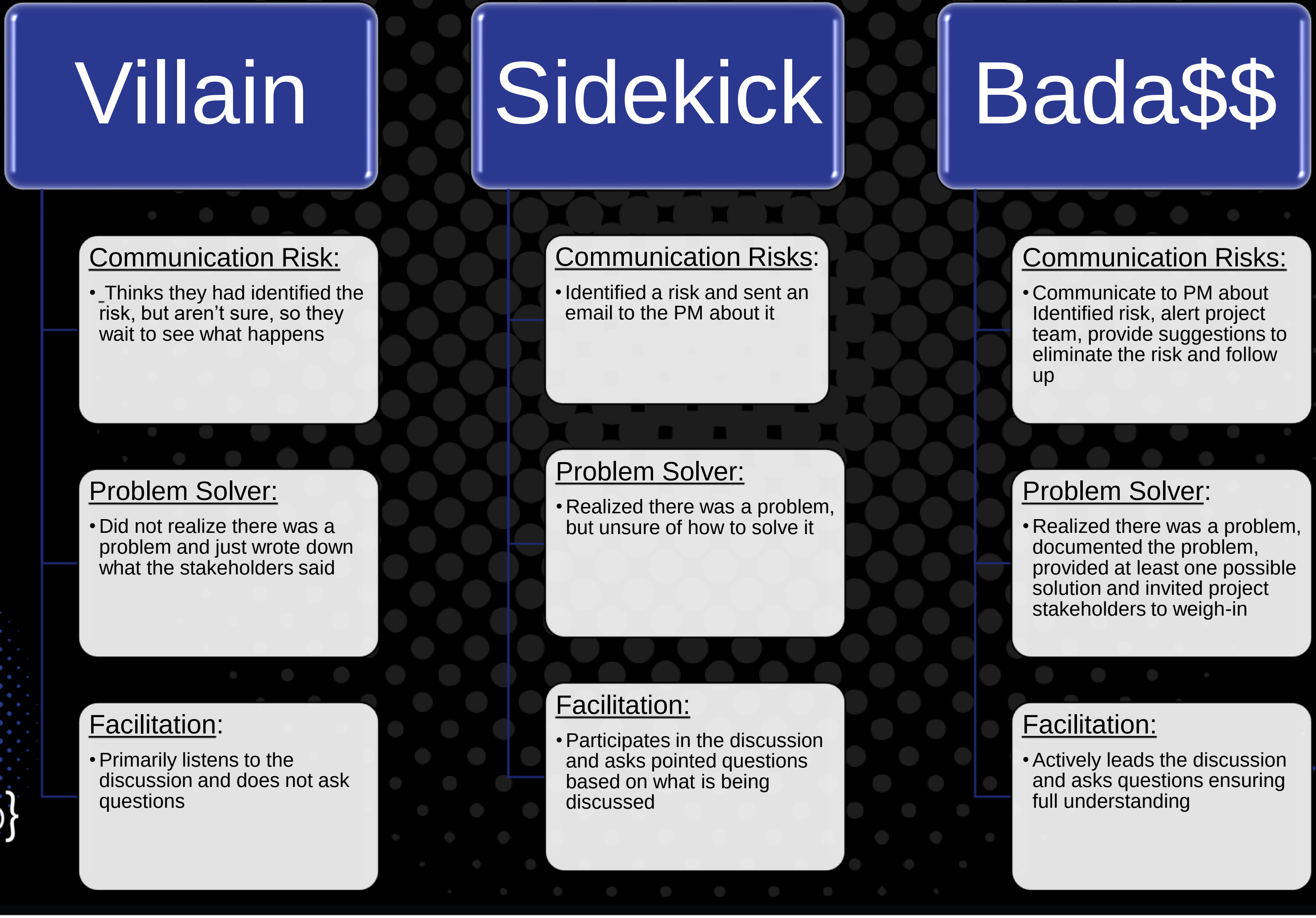
- Think holistically
- Define root cause of the problem
- Ask the same question twice
- Define acceptance criteria

BA Superpowers

★ Smooth Meeting Facilitation with Clients

- Prepare, Prepare, Prepare
- POW!
  - Purpose
  - Output
  - WIIFT
- Reiterate Goals

Superpower Status



Villain

Communication Risk:

- _Thinks they had identified the risk, but aren't sure, so they wait to see what happens

Problem Solver:

- Did not realize there was a problem and just wrote down what the stakeholders said

Facilitation:

- Primarily listens to the discussion and does not ask questions

Sidekick

Communication Risks:

- Identified a risk and sent an email to the PM about it

Problem Solver:

- Realized there was a problem, but unsure of how to solve it

Facilitation:

- Participates in the discussion and asks pointed questions based on what is being discussed

Bada$$

Communication Risks:

- Communicate to PM about Identified risk, alert project team, provide suggestions to eliminate the risk and follow up

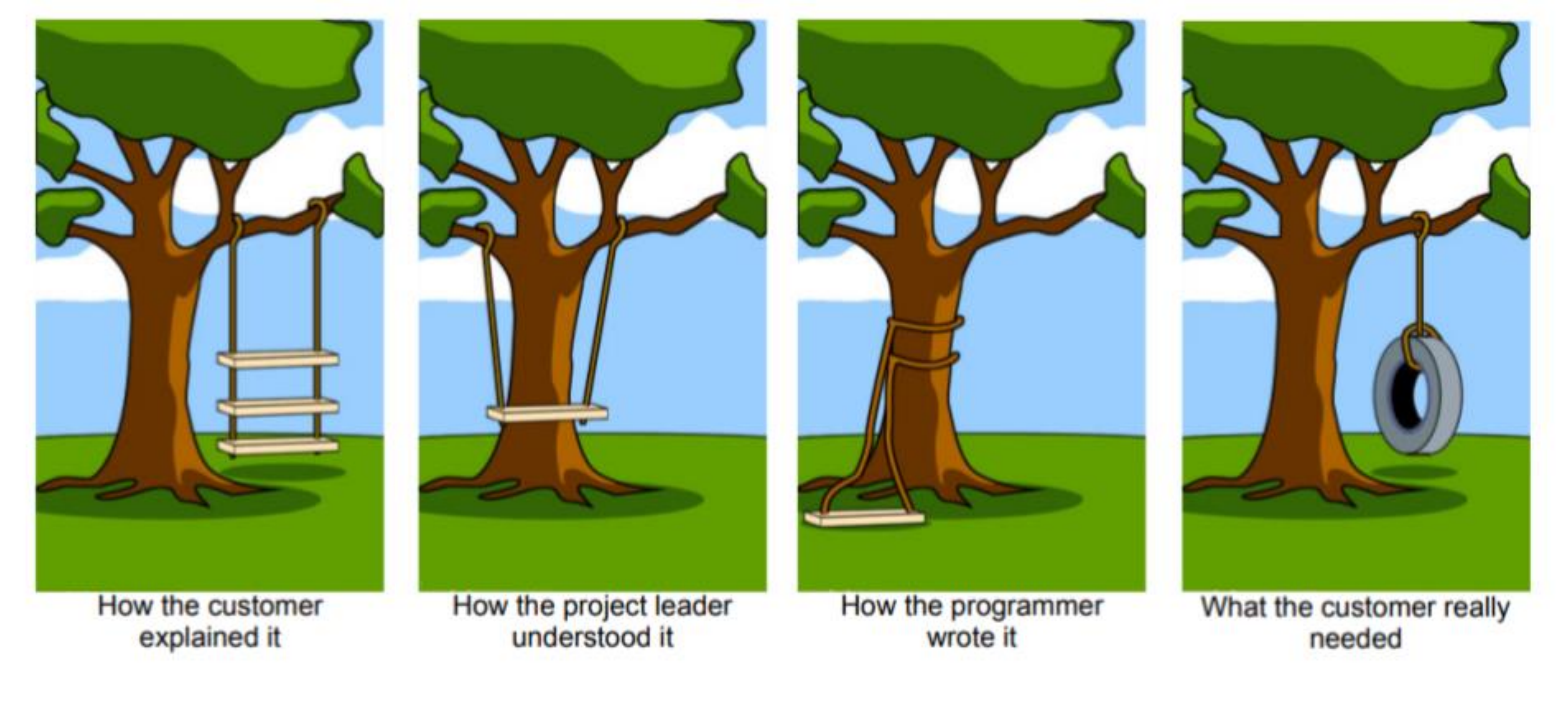
Problem Solver:

- Realized there was a problem, documented the problem, provided at least one possible solution and invited project stakeholders to weigh-in

Facilitation:

- Actively leads the discussion and asks questions ensuring full understanding

BAs and Requirements


How the customer explained it
How the project leader understood it
How the programmer wrote it
What the customer really needed

Requirements – what do we want to understand?

People

- Who are the people doing the work?
- Who are all the people they collaborate with to get their work done?

Process

- What is it that they do today (As-Is)?
- What is it that they want to do (To-Be)?
- What is the goal of the process?

Technology

- What tools do they use to get their job done?
- How does technology support what they do today?
- How does it NOT support what they do today?

What does a good requirement look like?

Think of these characteristics as a series of filters. A good requirement will pass all eight filters.

- Verifiable
- Clear & Concise
- Complete
- Consistent
- Traceable
- Viable
- Necessary
- Implementation Free

Verifiable

A **verifiable** requirement…

- Is stated in such a way that it **can be tested** by:
  - Inspection
  - Analysis
  - Demonstration
- Makes it possible to evaluate whether the system met the requirement

- Verifiable
- Clear & Concise
- Complete
- Consistent
- Traceable
- Viable
- Necessary
- Implementation Free

Bad Example:
The system must be user friendly.

Good Example:
As any system user, I need my interface to be menu driven and provide dialog boxes, help screens, radio buttons and dropdown list boxes to help me to do user inputs, so that I can easily navigate the screen.

Clear & Concise

A **clear & concise** requirement…

- Must consist of a single requirement
- Must be easily read and understood by non-technical people
- Must be unambiguous and not susceptible to multiple interpretations
- Must avoid subjective or open-ended terms

Bad Example:
All screens must appear on the monitor quickly.

Good Example:
As a Sales User, I must have the Opportunity report load within 2 seconds.

- Verifiable
- **Clear & Concise**
- Complete
- Consistent
- Traceable
- Viable
- Necessary
- Implementation Free

Complete

A **complete** requirement…

- Contains all the information that is needed to define the system function
- Leaves no one guessing (for how long?, 50% of what?)
- Includes measurement units (when appropriate)

- Verifiable
- Clear & Concise
- Complete
- Consistent
- Traceable
- Viable
- Necessary
- Implementation Free

Bad Example:
Must have offline capabilities.

Good Example:
As a mobile technician, I need a tool that can still allow me to access all Work Order functions when I am not able to access cellular service, so that I can service our remote customers.

Innovia CONSULTING

INGRAM MICRO CLOUD

SPS COMMERCE

Own {backup}

DYNAMICSCON LIVE

DUG

Consistent

A **consistent** requirement…

- Does not conflict with other requirements
- Uses the same terminology throughout the specification
- Does not duplicate other requirements or pieces of requirements or create redundancy in any way

- Verifiable
- Clear & Concise
- Complete
- **Consistent**
- Traceable
- Viable
- Necessary
- Implementation Free

Bad Example:
Requirement 1:
Opportunity should have Estimated Revenue
Requirement 2:
Estimated Revenue has to be reported on

Good Example:
As a salesperson, I should have the ability to capture Estimated Revenue related to any given Opportunity, so that I can measure my pipeline with reporting

Traceable

A **traceable** requirement…

- Has a unique identity or number
- Cannot be separated or broken down into smaller requirements
- Can be easily traced through to specification, design, and testing

Bad Example:
They need a lost opportunities report filtered by region and a lost opportunities report broken out by Salesperson

Good Example:
Requirement 1: As a sales manager, I need to have a report that shows me lost Opportunities by Region, so that I can measure all losses
Requirement 2: As a sales manager, I need to have a report that shows me lost Opportunities by salesperson, so that I can measure all losses

- Verifiable
- Clear & Concise
- Complete
- Consistent
- Traceable
- Viable
- Necessary
- Implementation Free

Viable

A viable requirement…

- Can be met using existing or planned technology
- Can be achieved within the budget and schedule
- Will be used by the end users
- Must be helpful to build the system

- Verifiable
- Clear & Concise
- Complete
- Consistent
- Traceable
- Viable
- Necessary
- Implementation Free

Bad Example:
The new system should be installed with no disruption to customer service whatsoever.

Good Example:
As a customer service rep, I should not experience more than 4 hours of downtime upon installation of the new Case functionality, so that I can easily pick up my in-flight Cases where I left off.

Necessary

A **necessary** requirement…

- Is one that must be present to meet system objectives
- Is absolutely critical for the operation of the system or process
- Leads to a deficiency in the system or process if it is removed

Bad Example:
Users should be able to log into the system.

Good Example:
As a Sales Manager, I need to have a security role that will allow me to access the Sales Leadership Model-Driven-App, so that I can view all of the processes and functionality required by my role.

- Verifiable
- Clear & Concise
- Complete
- Consistent
- Traceable
- Viable
- Necessary
- Implementation Free

Innovia CONSULTING

INGRAM MICRO CLOUD

SPS COMMERCE

Own{backup}

DYNAMICSCON LIVE

Implementation Free

A requirement is **implementation free** when …

- It defines what functions are provided by the system
- Does not specify how a function can or should be implemented
- Allows the system developer to decide what technology or method is best suited to achieve the function

Bad Example:
After clicking a button in the command bar, Java Script should run and kick off a notification to the user with a message saying “Operation Successful”.

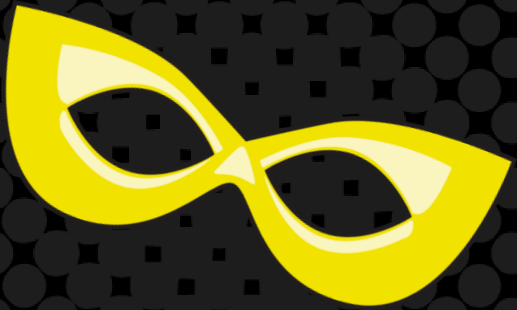
Good Example:
As a Salesperson, when I am ready to generate my Order Connection to F&O, I should be able to take an action and get a notification or signal, so that I know that my operation was successful.

- Verifiable
- Clear & Concise
- Complete
- Consistent
- Traceable
- Viable
- Necessary
- **Implementation Free**

Innovia Consulting
Ingram Micro Cloud
SPS Commerce
Own {backup}

DynamicsCon Live

What tip would you suggest?



Tips for Creating Bada$$ Requirements

Ask “Stupid” Questions

- Start with what you already know
- Make a solid suggestion
- Ask with confidence
- Pretend like you are a new employee (because you are!)
- Don’t ask Yes/No Questions
- Always ask “Why?”

Tips for Creating Bada$$ Requirements

Form Your Squad & Follow-up

- Use "we" early on to facilitate team vibes
- Build the requirements collaboratively
- Hold regular review meetings with business and IT
- Define tasks and provide updates

Tips for Creating Bada$$ Requirements

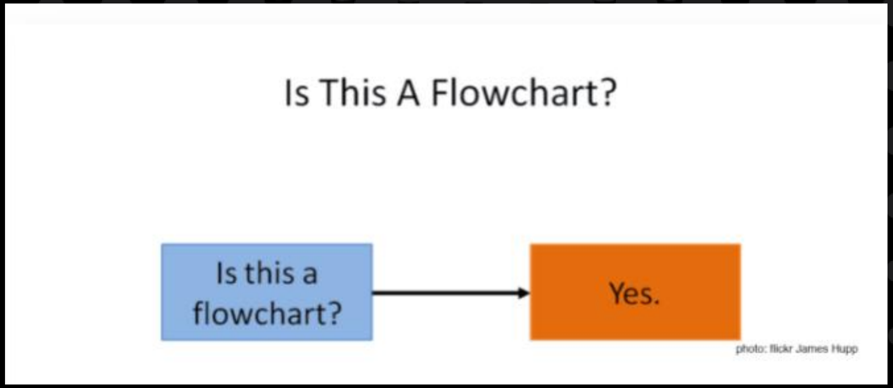
Be Patient, but Firm

- Do not get frustrated with changing requirements
- Remember – it's an iterative process
- Obtain team approval for requirements

Tips for Creating Bada$$ Requirements

Focus, Listen, Document

- Resist solutioning during requirements gathering
- Ask a question. Listen to understand. Repeat it back.
- Keep requirements documentation organized and available
- Flex your creative skills with flow charts or use case diagrams



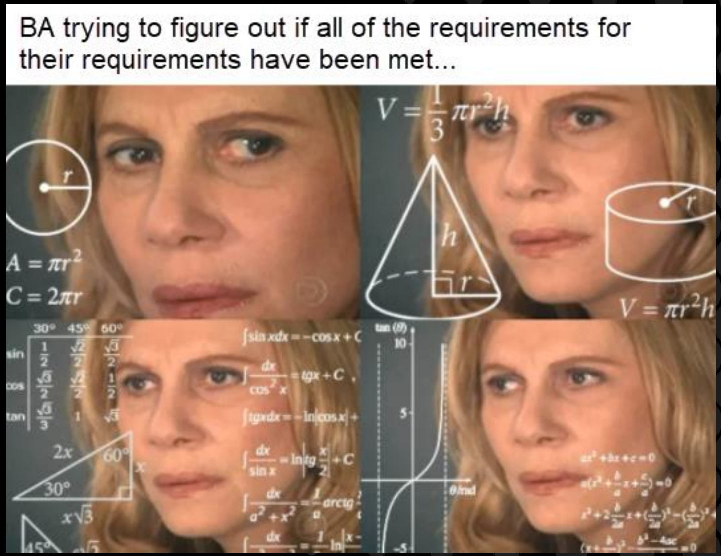
Is This A Flowchart?

Is this a flowchart?

Yes.

photo: flickr James Hupp

Tips for Creating Bada$$ Requirements



BA trying to figure out if all of the requirements for their requirements have been met...

Check Yourself

- Do the requirements make sense?
- Could anyone reading this understand it?
- Does it pass all the filters of a good requirement?
- Have you collaborated on this requirement with the client?
- Does this need a visual to paint a better picture?
- Is this requirement thinking about the future state?
- Are the requirements well organized and visible to the client?

Innovia CONSULTING

INGRAM MICRO CLOUD

SPS COMMERCE

Own {backup}

DYNAMICSCON LIVE

Questions?

ALL IN A DAYS WORK
FOR YOUR FRIENDLY NEIGHBOURHOOD BUSINESS ANALYST